



# Modern Block Cipher



# Outline



- Modern Symmetric Key Cipher
- Modern Block Cipher
- Block Cipher as Permutation Groups
- Components of a Modern Block Cipher
- Diffusion vs Confusion
- Classes of Product Cipher
- Modern Stream Cipher

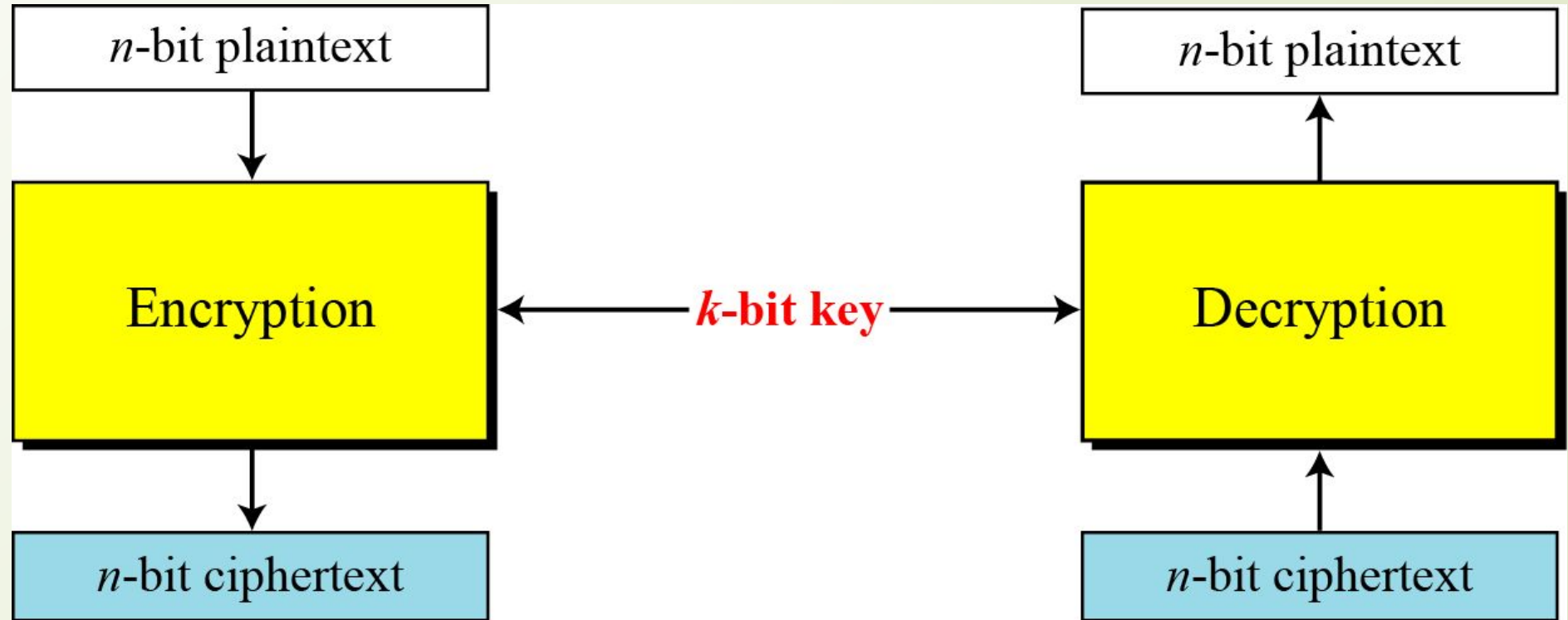
# Modern Symmetric- Key Cipher

- Traditional Cipher are **Character Oriented Ciphers**.

Stream Cipher	Block Cipher
Additive Cipher Monoalphabetic Cipher Vigenere Cipher	Playfair Cipher Hill Cipher

- As Information consist of **Numbers, Graphics, Audio, Video** etc., We require **bit-oriented ciphers**.

# Modern Block Cipher





# Continue...

- ❑ **Symmetric Key Modern Block Cipher:**
  - ❑ Encrypts an  $n$ -bit block of plaintext
  - ❑ Decrypts an  $n$ -bit block of Cipher text
- ❑ Key size will be  $n$ -bit for both encryption and decryption.
- ❑ The Decryption algorithm must be the inverse of an Encryption algorithm.



## Continue...

- If **the message** is **smaller than n-bit** or has fewer than n-bits, **Padding** must be added to make it an n-bit block.
- If the message has more than n-bits, it should be divided into n-bit blocks and the appropriate padding must be added to the last block of the message.

# Example

How many padding bits must be added to a message of 100 characters if 8-bit ASCII is used for encoding and the block cipher accepts blocks of 64 bits?

## Solution

- Message length=**100** characters
- Message length in bit=  $100 * 8 = \mathbf{800 \text{ bit}}$
- Block Size=**64**

$$\begin{aligned}\text{Encoded msg} &= (\text{Message} + \text{Padding}) \\ 64 &= (100*8) + \text{Padding} \\ (64 - 800) \bmod 64 &= \text{Padding}\end{aligned}$$

So 32 bits of padding is required.



# Substitution or transposition

- ❑ **Substitution Cipher** : a 1-bit or 0-bit in the plaintext can be replaced by either a 0 or a 1
  - ❑ This means that P.T & C.T can have different number of 1's or 0's
  - ❑ Number of n-bit possible P.T & C.T is  $2^n$
- ❑ **Transposition Cipher** : bits are only reordered
  - ❑ Same number of 1's or 0's in P.T & C.T
  - ❑ Number of n-bit possible P.T & C.T is  $(n!) / (\text{no of 1's !}) \times (\text{no of 0's !})$





# Example

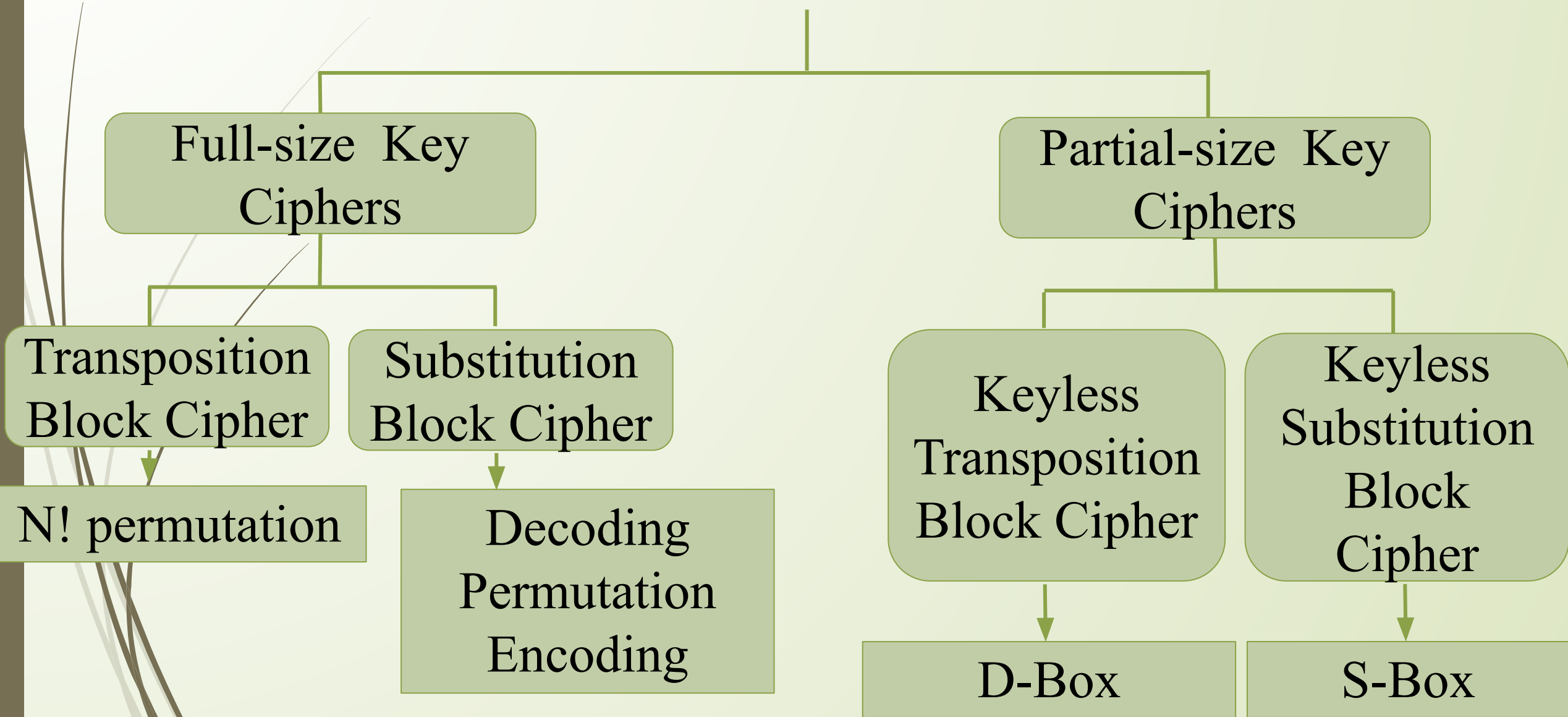
- Suppose that we have a block cipher where  $n = 64$ . If there are 10 1's in the ciphertext, how many trial-and-error tests does Eve need to do to recover the plaintext from the intercepted ciphertext in each of the following cases?



# Solution

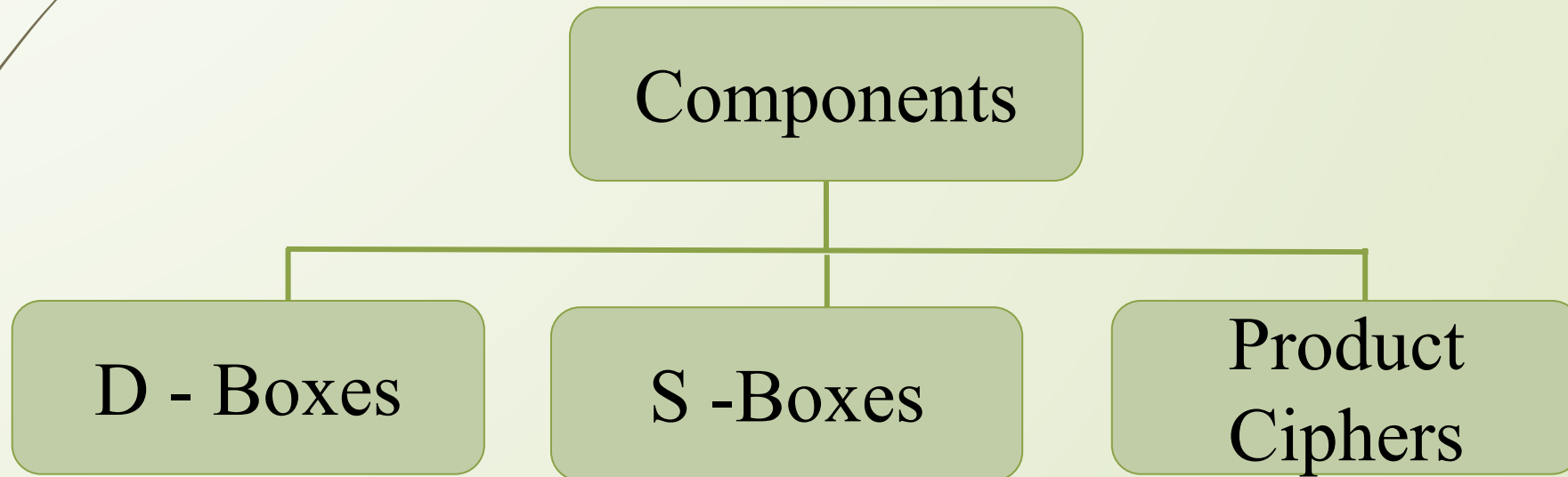
- ❑ The cipher is designed as a substitution cipher
- ❑ The cipher is designed as a transposition cipher
- ❑ Substitution Cipher :
- ❑ Don't know about how many no of 0's or 1's are in the P.T so will try all the possible values that is  $2^{64}$
- ❑ Transposition Cipher :
- ❑ Knows that there is exact 10 1's in the P.T so that will try only those values which contains exactly 10 1's
- ❑  $64!/(10!)(54!)$

# Block Ciphers as permutation groups



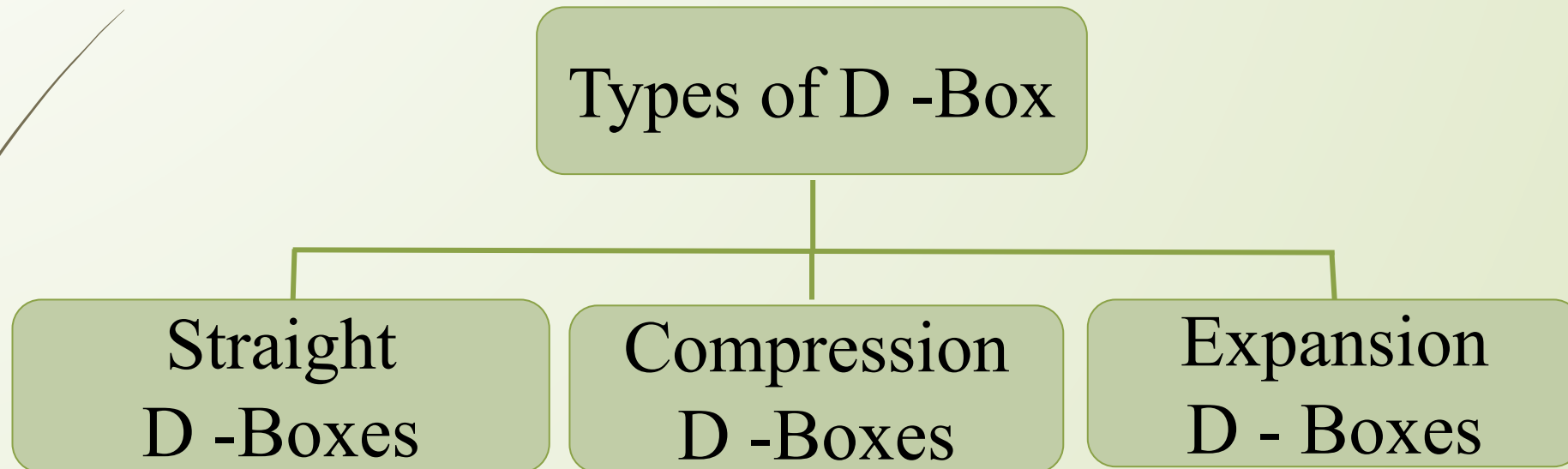
# Components of Modern block cipher

- Modern block ciphers are normally keyed substitution ciphers in which key allows only partial mapping from the possible inputs to possible outputs.



# D - Boxes

- Diffusion Box – transposes bits



# Straight D - Boxes


- With  $n$  inputs and  $n$  outputs is a permutation
- $N!$  possible mappings
- Keyless – mapping is predetermined

## Example of D - Box

58	50	42	34	26	18	10	02	60	52	44	36	28	20	12	04
62	54	46	38	30	22	14	06	64	56	48	40	32	24	16	08
57	49	41	33	25	17	09	01	59	51	43	35	27	19	11	03
61	53	45	37	29	21	13	05	63	55	47	39	31	23	15	07



## Example:

- Design an  $8 \times 8$  permutation table for a straight D-Box that moves the two middle bits in the input word to the two ends in the output words. Relative positions of other bits should not be changed.
- 



# Solution

□ Normal Permutation : [ 1 2 3 4 5 6 7 8 ]

□ Middle Bits are 4 & 5

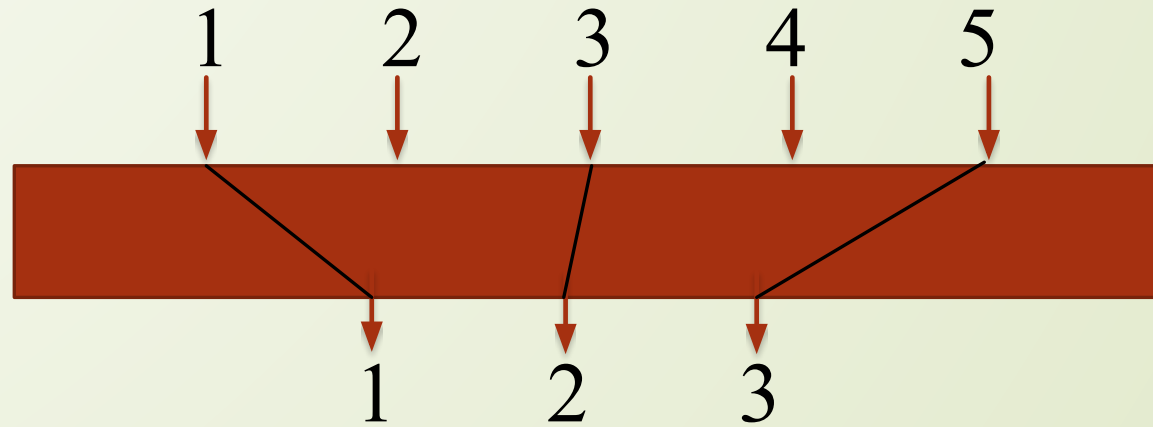
□ End bits are 1 & 8

□ D – Box : [ 4 1 2 3 6 7 8 5 ]



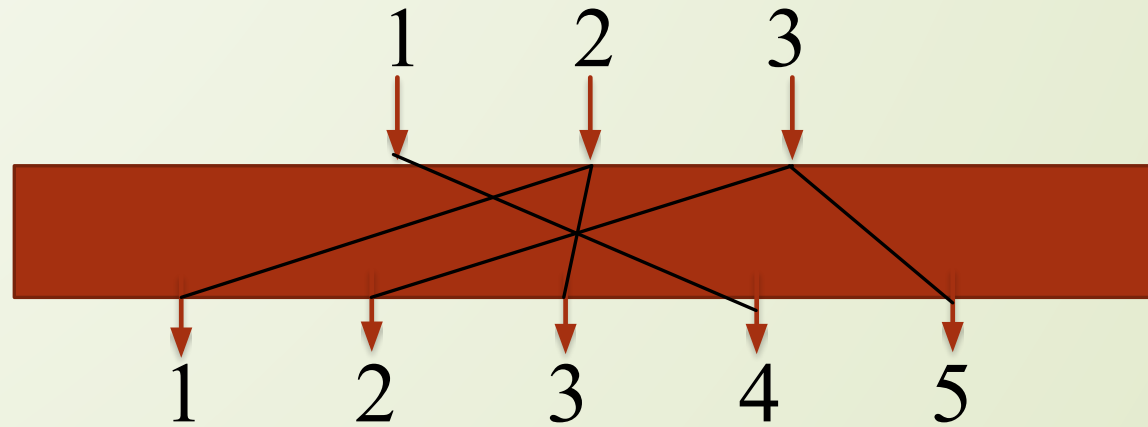
# Compression D - Boxes

- With  $n$  inputs and  $m$  outputs , where  $m < n$
- Used when we need to permute bits and decrease no of bits for the next stage
- Keyless – mapping is predetermined



# Expansion D - Boxes

- With  $n$  inputs and  $m$  outputs , where  $m > n$
- $m - n$  entries will be repeated
- Used when we need to permute bits and increase no of bits for the next stage



Continue .....

## Inevitability: Only Straight D box is invertible

□ Original Table :	6	3	4	5	2	1
□ Add indices :	1	2	3	4	5	6
□ Swap Content :	1	2	3	4	5	6
□	6	3	4	5	2	1
□ Sort indices :	6	5	2	3	4	1
	1	2	3	4	5	6

# S - Boxes

- Substitution Box
- N bit inputs and m bit outputs where m and n are not necessarily the same
- Can be keyless or keyed
- If  $n = m$  then and only then S box is invertible

Encryption side S Box

	00	01	10	11
0	011	101	111	100
1	000	010	001	110

Decryption side S Box

	00	01	10	11
0	100	110	101	000
1	011	001	111	010



# Product Cipher

- **A Product cipher** combines **two or more transformations** in a manner intending that the **resulting cipher is more secure** than the individual components to make it resistant to cryptanalysis.
- Product Cipher is a complex cipher combining **substitution, permutation, and other components** of modern block cipher.



# Continue...

## Product Cipher



```
graph TD; A[Product Cipher] --> B[Diffusion]; A --> C[Confusion]; B --> D[Hide relationship between the ciphertext and the plaintext]; B --> E[Implies that each symbol in the ciphertext is dependent on some or all symbols in the plaintext]; C --> F[Hide relationship between ciphertext and key]; C --> G[If a single bit in the key is changed, most or all bits in ciphertext will also be changed];
```

### Diffusion


- ❑ Hide relationship between the ciphertext and the plaintext
- ❑ Implies that each symbol in the ciphertext is dependent on some or all symbols in the plaintext

### Confusion

- ❑ Hide relationship between ciphertext and key
- ❑ If a single bit in the key is changed, most or all bits in ciphertext will also be changed

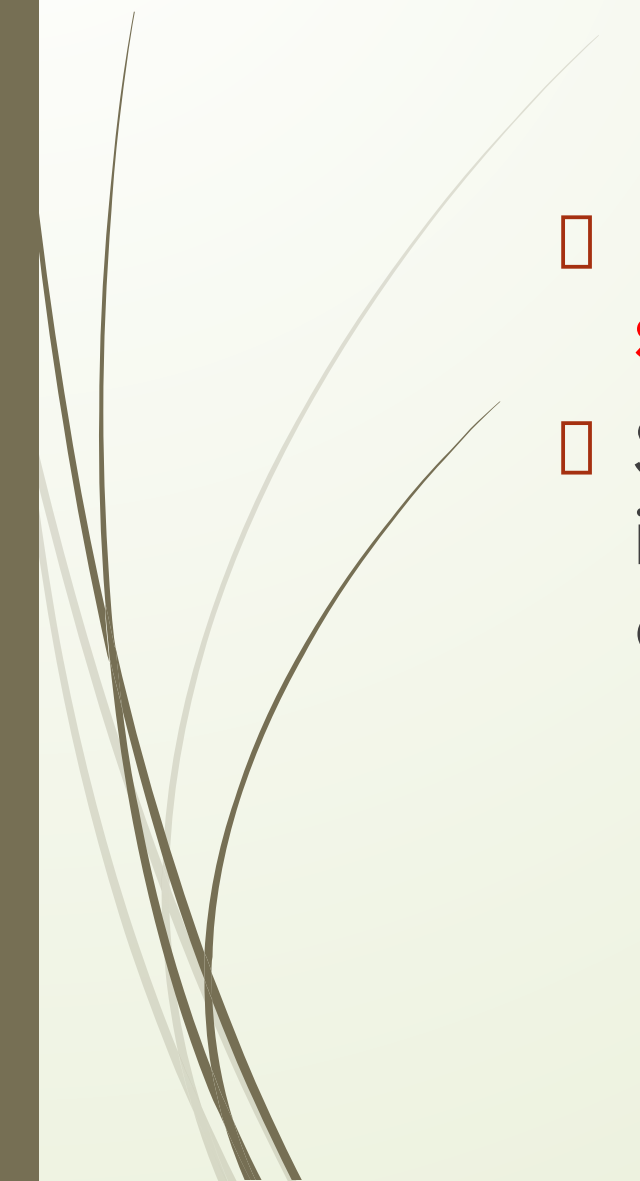


# Classes of Product Cipher

1. **Feistel Cipher** : uses both invertible and non invertible components
  2. **Non-Feistel Cipher** : only uses invertible components.
- 

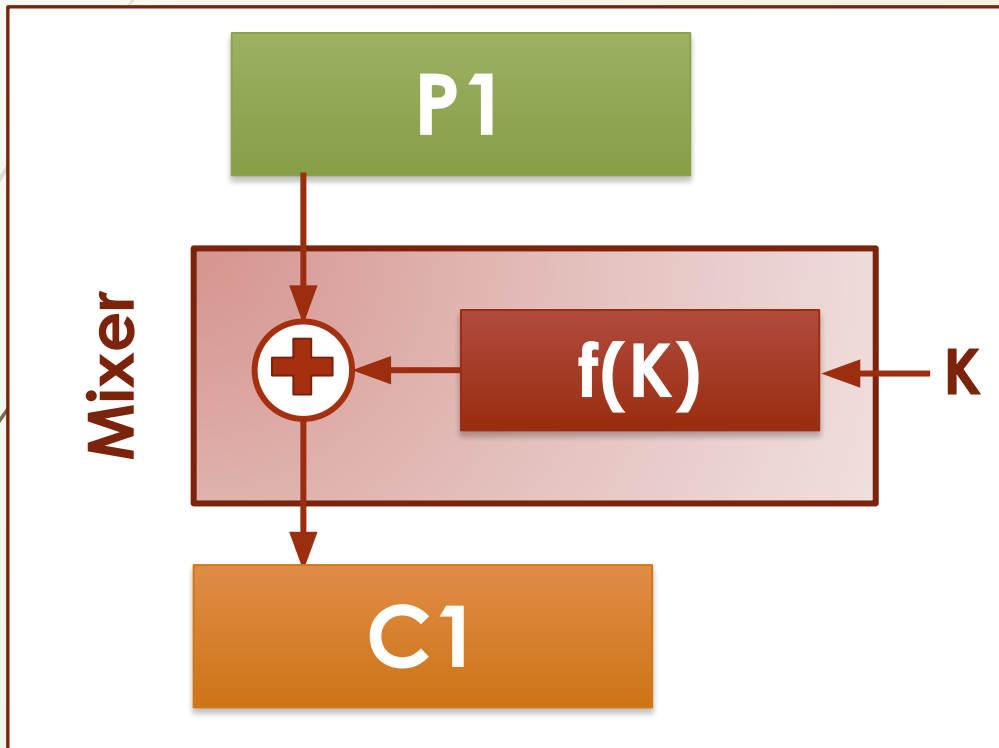


# Feistel Cipher

- Feistel Cipher can have **3 types of components: self-invertible, invertible and non-invertible.**
  - Self inverse means that the function is its own inverse: if you apply it twice, you get back your original input.
- 

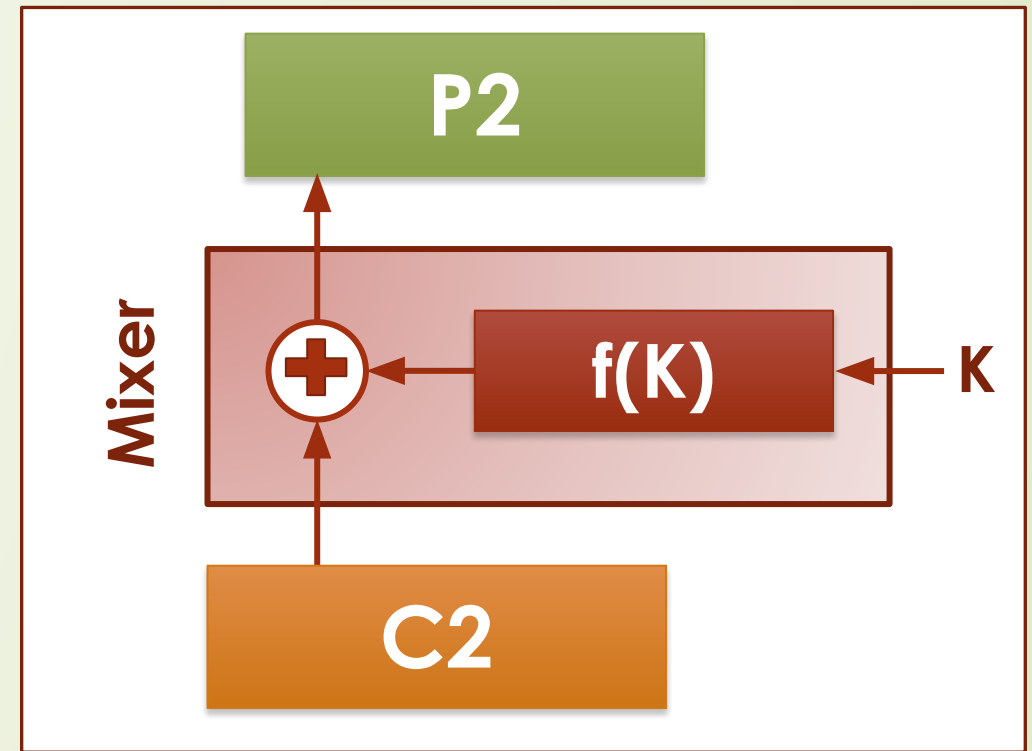


## First Thought



**Encryption:**

$$C1 = P1 \oplus f(K)$$



**Decryption:**

$$\begin{aligned} P2 &= C2 \oplus f(k) \\ &= C1 \oplus f(K) \\ &= P1 \oplus f(K) \oplus f(K) \\ &= P1 \oplus (000\dots 00) \\ &= P1 \end{aligned}$$



## Example:

- The plaintext and cipher text : 4 bits long and
- the key: 3 bits long.
- The function takes the first and third bits of the key, interprets these two bits as a decimal number, squares the number, and interprets the result as a 4-bit binary pattern.
- Show the results of encryption and decryption if the original plaintext is 0111 and the key is 101.

# Solution

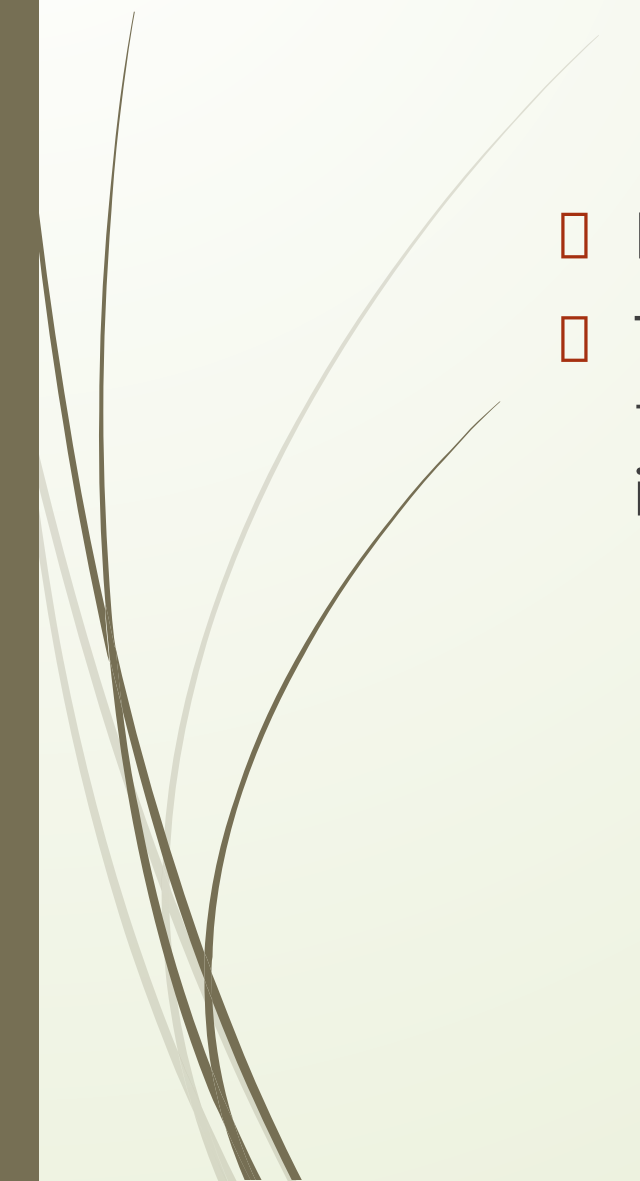
- The function extracts the first and second bits to get **11** in binary or **3** in decimal.
- The result of squaring is **9**, which is **1001** in binary.

**Encryption:**  $C = P \oplus f(K) = 0111 \oplus 1001 = 1110$

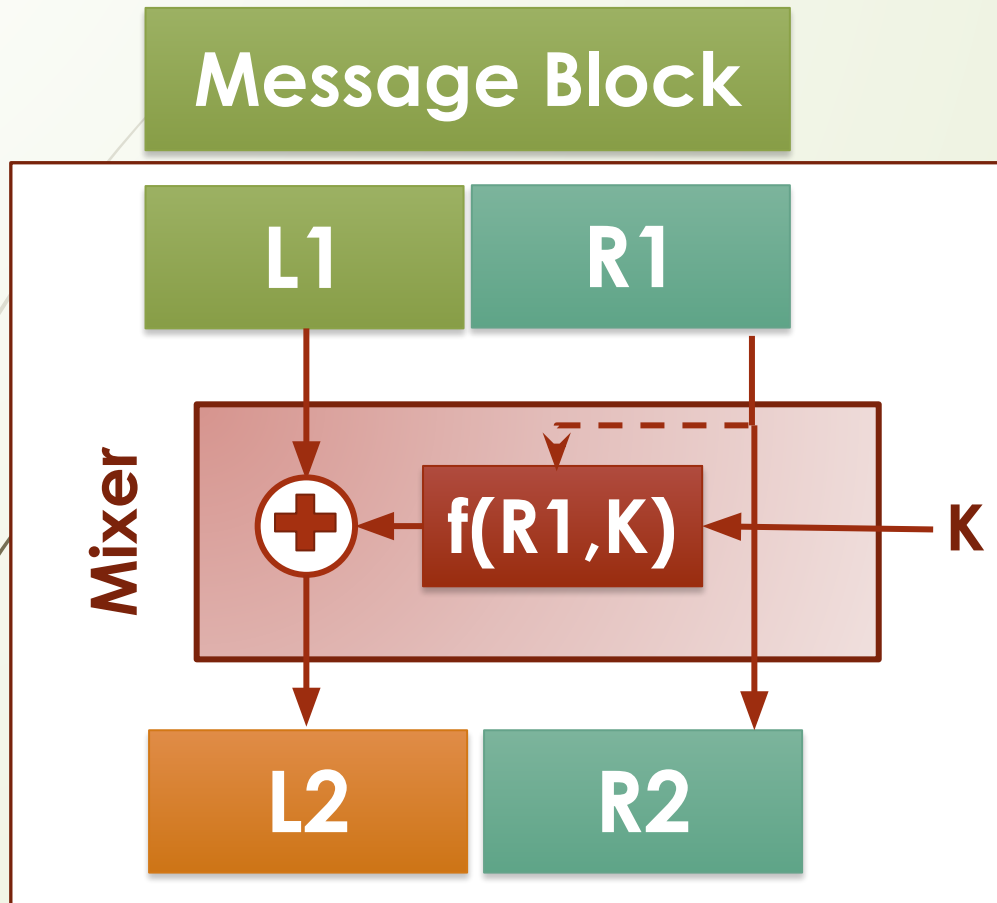
**Decryption:**  $P = C \oplus f(K) = 1110 \oplus 1001 = 0111$



# Limitations

- ❑ Here Function is accepting only single input that IS KEY.
  - ❑ To make it more complex and secure, Plaintext must be the input to the function and key must be the second input.
- 

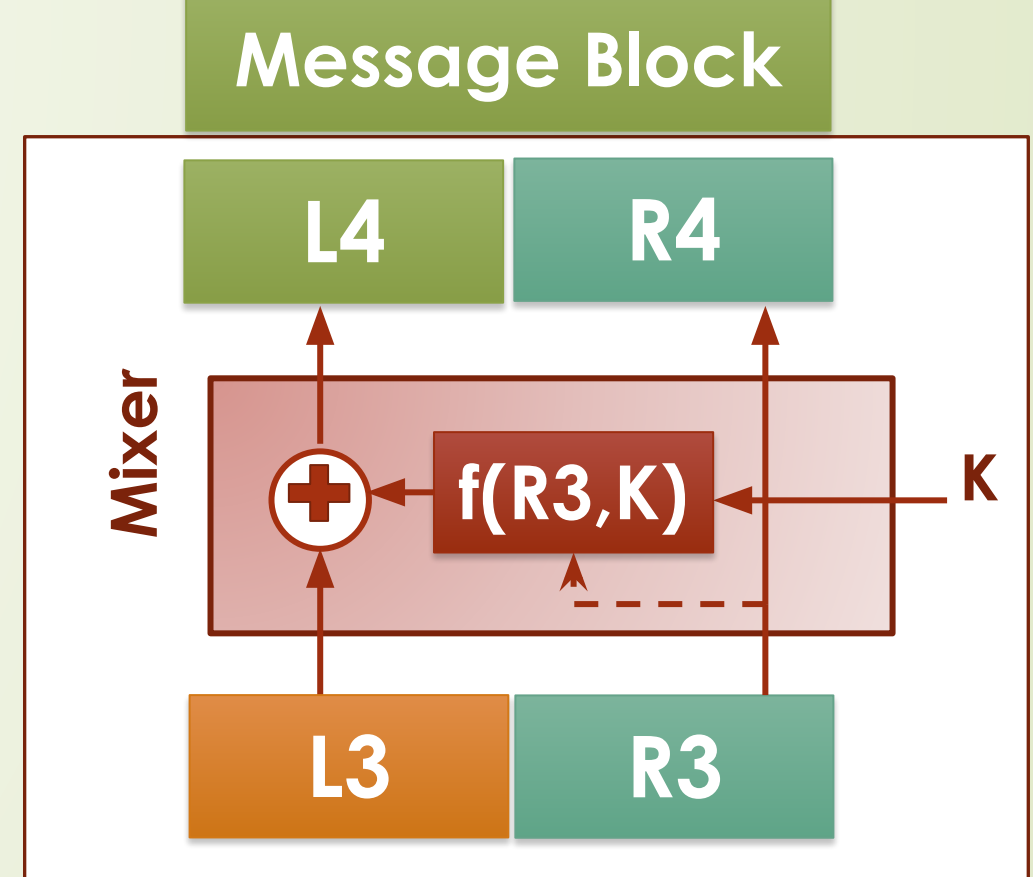
# Improvement



**Encryption:**

$$R2 = R1$$

$$L2 = L1 \oplus f(R1, K)$$



**Decryption:**

$$L4 = L3 \oplus f(R3, k)$$

$$= L2 \oplus f(R2, K)$$

$$= L1 \oplus f(R1, K) \oplus f(R1, K)$$

$$= L1 \oplus (000\dots 00)$$

$$= L1$$



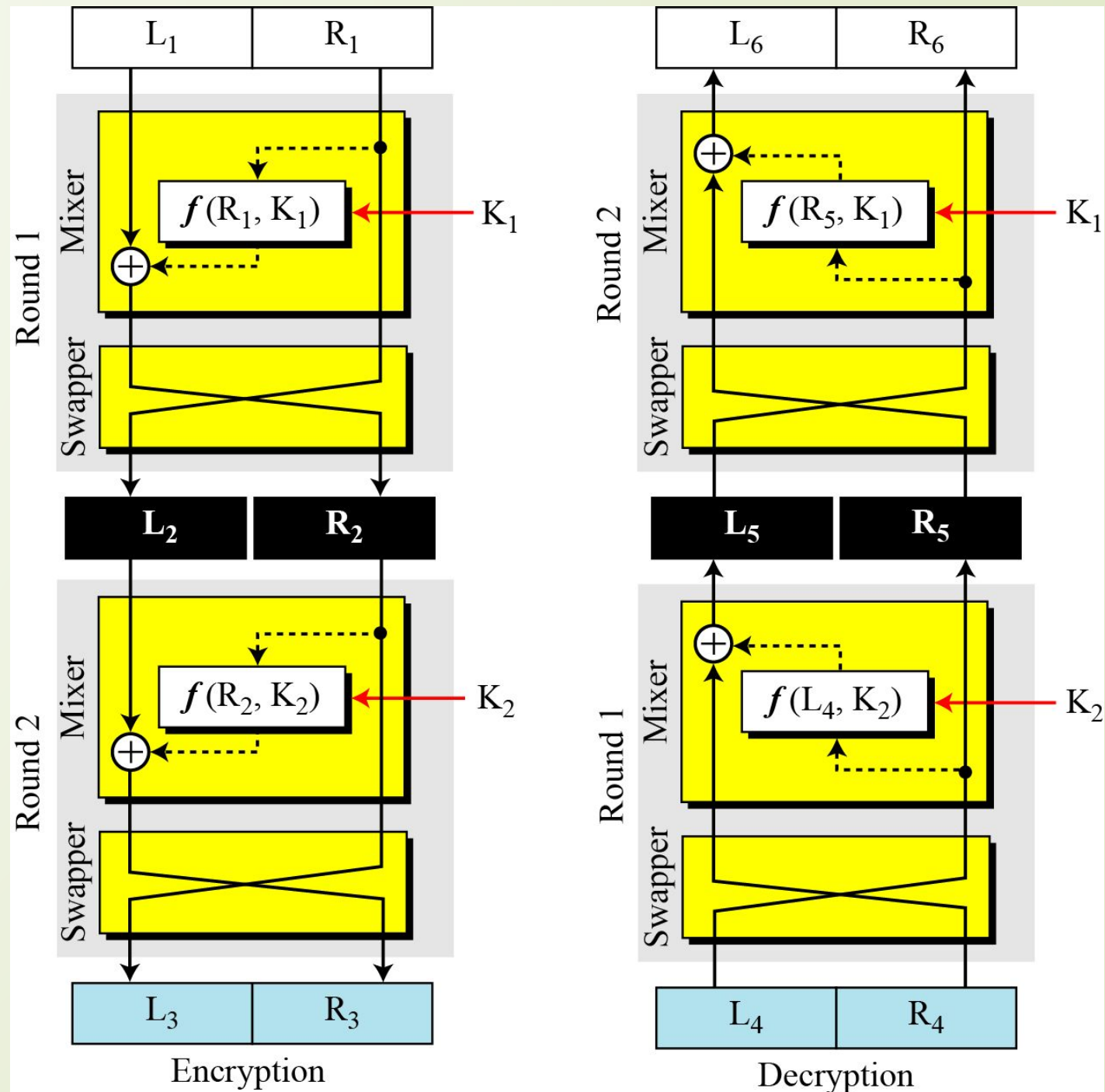
# Limitations

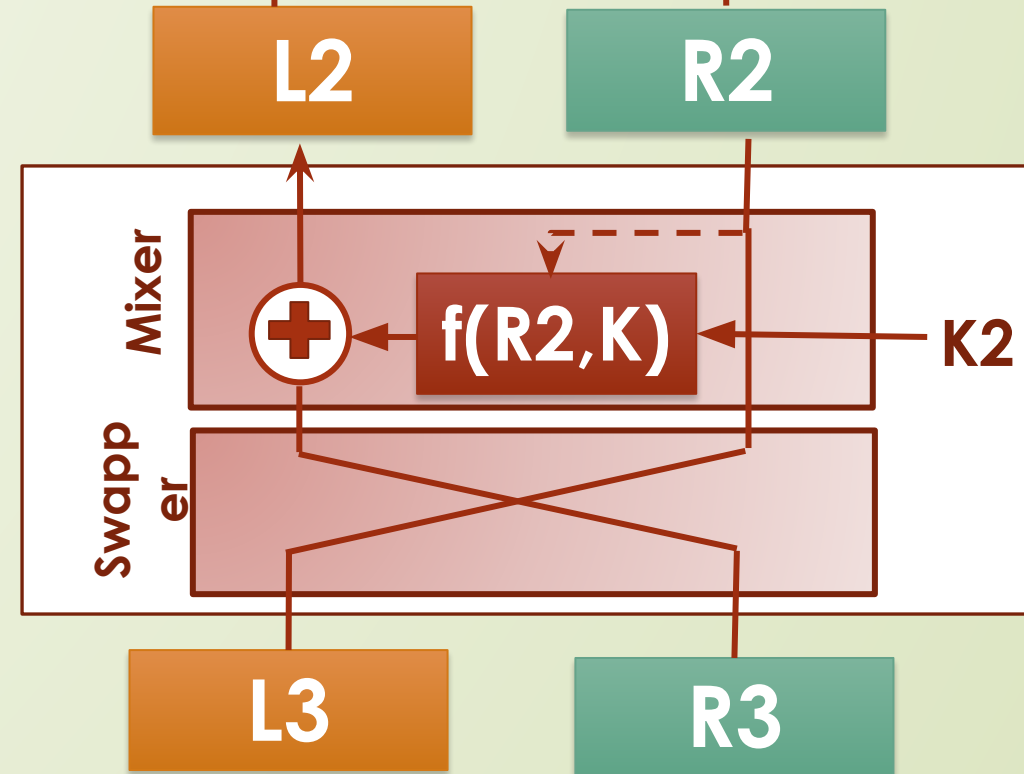
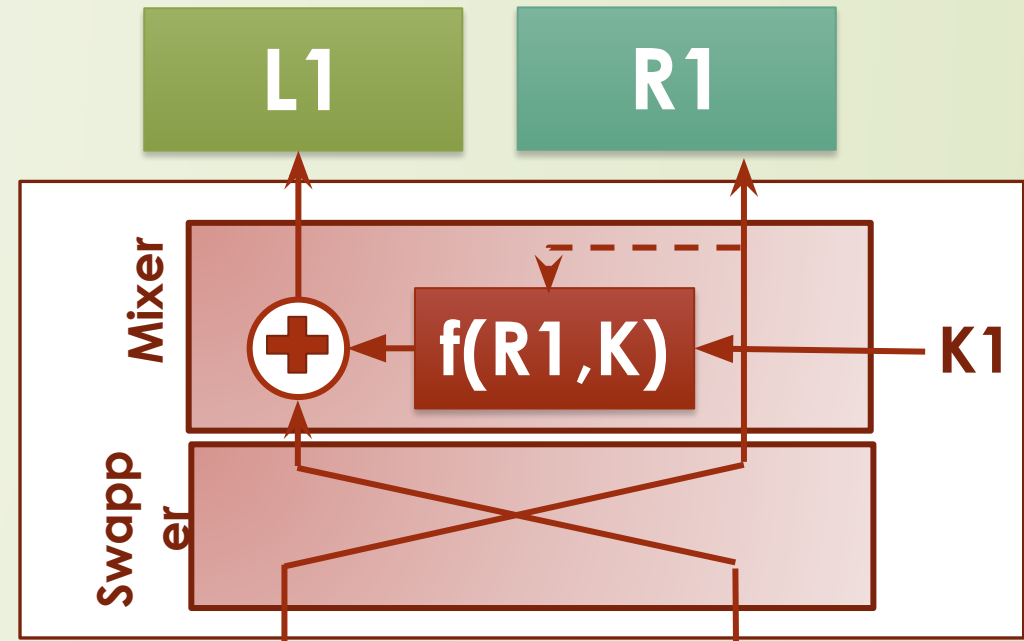
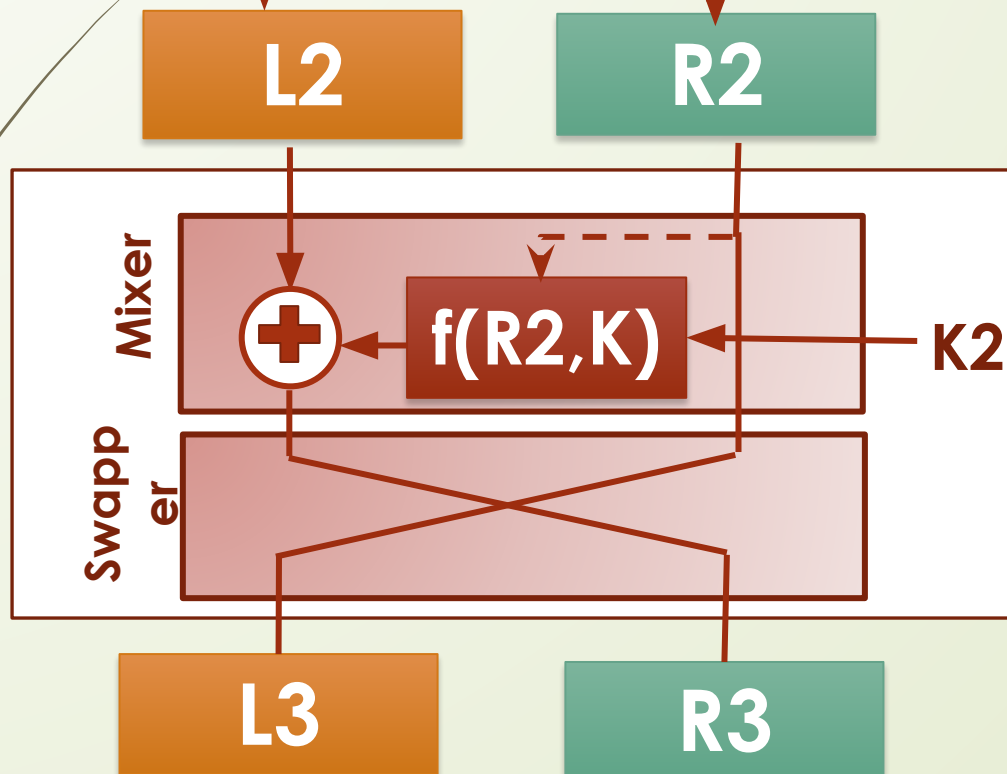
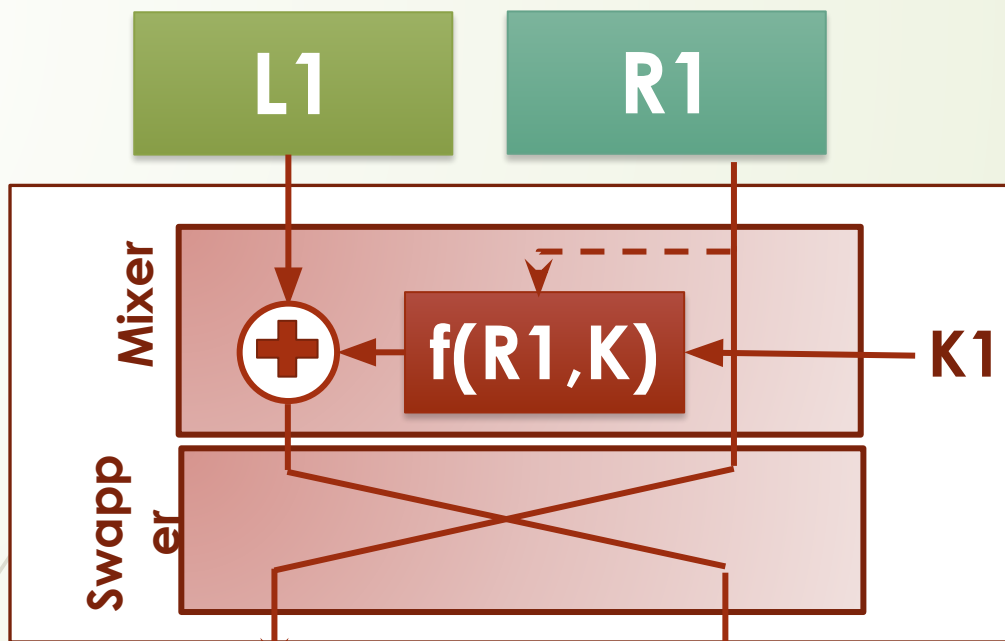
- ❑ Here  $R1=R2=R3=R4$  which is never getting changed.
- ❑ Eve can immediately find the right half of the plaintext by intercepting the cipher text and extracting the right half of it.

# Improvements

- ❑ increase no of rounds.
- ❑ Add new Element: **swapper**

# Final design of a Feistel cipher with two rounds



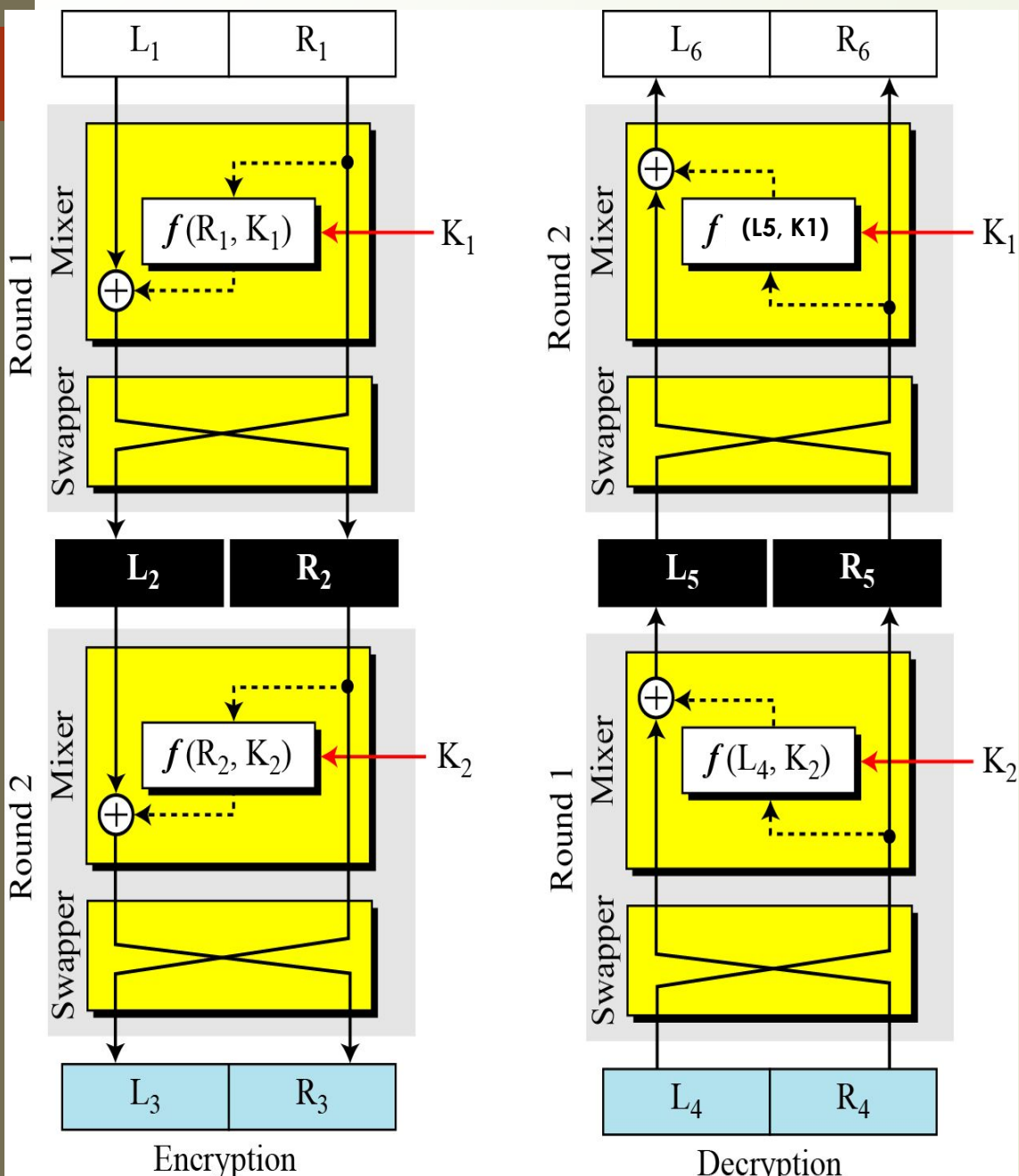






## Continue..

- The effect of Swapper in the encryption will be canceled by the effect of the swapper in the decryption.
- The keys will be used in the reverse order in the encryption and decryption.
- As swapper are inverse of each other and mixer are inverse of each other, encryption and decryption will be inverse of each other.



□ Assume  $L_3=L_4$  and  $R_4= R_3$

$$\begin{aligned}
 \square L_5 &= R_4 \oplus f(L_4, K_2) \\
 &= R_3 \oplus f(R_2, K_2) \quad (\text{As } L_4=L_3=R_2) \\
 &= L_2 \oplus f(R_2, K_2) \oplus f(R_2, K_2) \\
 &= L_2
 \end{aligned}$$

$$R_5=L_4=L_3=R_2$$

$$\begin{aligned}
 \square L_6 &= R_5 \oplus f(L_5, K_1) \\
 &= R_2 \oplus f(L_2, K_1) \\
 &= L_1 \oplus f(R_1, K_1) \oplus f(R_1, K_1) \\
 &= L_1
 \end{aligned}$$

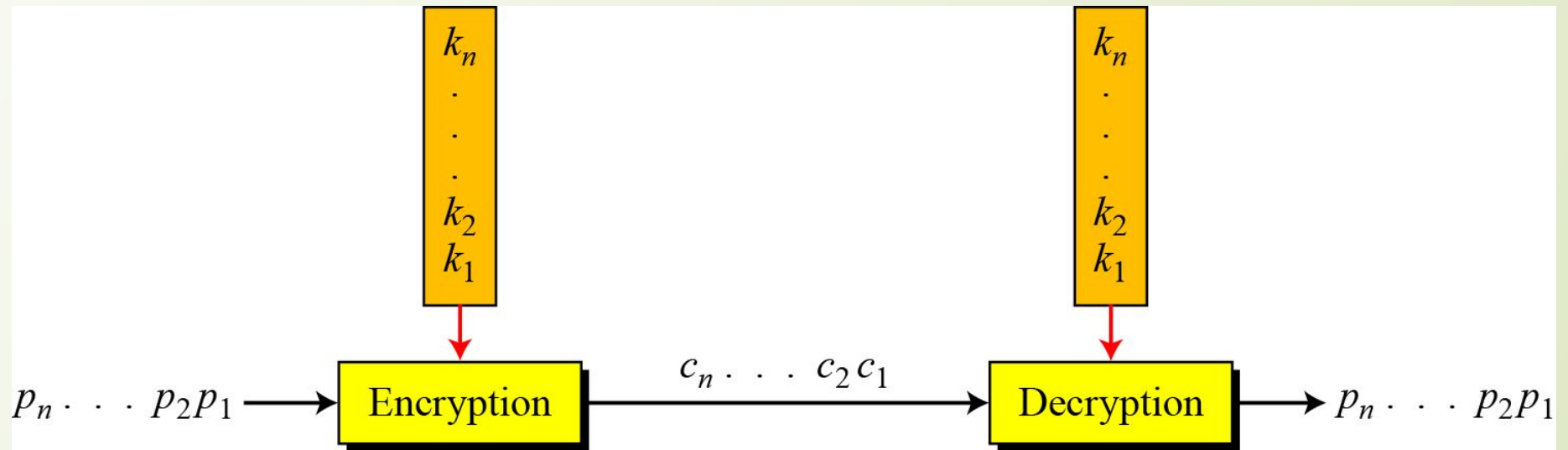
$$R_6=L_5=L_2=R_1$$



# Modern Stream Cipher

- In a modern stream cipher, encryption and decryption are done  $r$  bits at a time.
- We have a plaintext bit stream  $P = p_n \dots p_2 p_1$ , a cipher text bit stream  $C = c_n \dots c_2 c_1$ , and a key bit stream  $K = k_n \dots k_2 k_1$ , in which  $p_i$ ,  $c_i$ , and  $k_i$  are  $r$ -bit words.
-

# Stream Cipher





□  $C_i = E(k_i, P_i)$

□  $P_i = D(k_i, C_i)$



# Continue...

- ❑ Stream Cipher are faster than block Ciphers.
  - ❑ Hardware Implementation of stream cipher is also easier.
  - ❑ When we need to encrypt the binary stream and transmit them at a constant rate, a stream cipher is the better choice to use.
  - ❑ Stream Ciphers are more immune to the corruption of bits during transmission.
- 



**In modern Stream Cipher, each  $r$ -bit word in the plain text stream is enciphered using an  $r$ -bit word in the key stream to create corresponding  $r$ -bit word in the cipher text stream.**

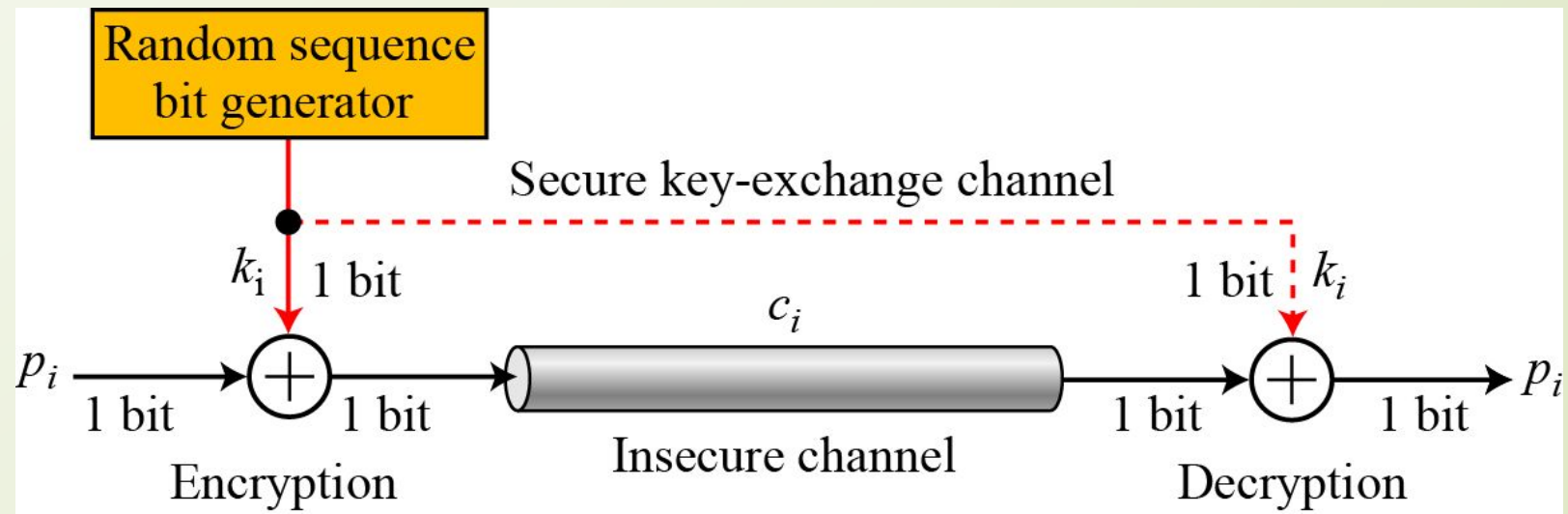


# Categories of Modern Stream Cipher

- Synchronous Cipher
  - Non synchronous Stream Cipher
- 

# Synchronous Stream Cipher

- In a synchronous stream cipher the key is independent of the plaintext or ciphertext.
- The Key stream is generated and used with no relationship between key bits and the plaintext or ciphertext bits.
- **Example; One time pad**







# References



- Cryptography and network security – Behrouz a forouzan,  
debdeep mukhopadhyay



# Thank You